



## Guardando las modificaciones

### Transcripción

[00:00] Hola a todos. Bienvenidos de nuevo a este curso de Git. En el video pasado lo que vimos fue cómo crear nuestro repositorio. Y ahora, por ejemplo, si nosotros colocamos `Git status` y enter ahí lo que vemos es que Git ya reconoce nuestra carpeta como repositorio pero está viendo que nuestros archivos aún no son trackeados o los cambios que hacemos a nuestros archivos no están siendo monitoreados por Git.

[00:28] Entonces lo primero que necesitamos hacer para guardar una modificación de un archivo en Git, es hacer que Git monitoree los cambios de ese archivo. Como no está siendo monitoreado ese archivo que es aquí el `index.html`, necesitamos usar el comando `git add`, espacio, y el nombre del archivo, `index.html` en este caso.

[00:56] Pero, ¿y si tuviera varios archivos? ¿Tengo que ir poniendo cada uno de los nombres de mis archivos? No. Con colocar `git add` punto, lo que hacemos es agregar todos los archivos que están dentro de esa carpeta y hacemos que Git monitoree todo esos archivos. En este caso voy a usar ese comando `git add` punto, presiono enter y cuando doy enter parece como si nada ocurrió.

[01:27] Pero si por ejemplo yo escribo `git status`, doy enter, acá vemos que ahora aparece en verde nuestro archivo. Esto significa que ahora Git ya está monitoreando ese archivo. Miren el mensaje. Cambios a ser commiteados o enviados o guardados. ¿Y si no queremos monitorear más ese archivo, este `index.html`? Lo que podemos hacer es usar este comando de acá: `"git rm --cached"` espacio, el nombre del archivo.

[02:03] Y con eso Git dejaría de monitorear ese archivo `index.html`. Pero eso no es lo que nosotros queremos hacer. Ahora lo que queremos hacer es guardar esa modificación en el repositorio de Git. Lo que hicimos es decirle a Git que monitoree ese archivo pero los cambios aún no han sido guardados.

[02:21] Entonces podemos crear un checkpoint o un punto dentro de la historia de ese archivo para decir: "Acá ocurrió alguna cosa en este archivo, alguna modificación." Y este punto se llama commit. En ese commit vamos a colocar nuestro archivo y un mensaje indicando lo que se modificó en ese archivo.

[02:42] Entonces, podemos hacer `git commit`, espacio. ¿Y `git commit` es suficiente para commitear, para guardar un archivo? No. Lo que necesitamos hacer es agregar un parámetro que es el `-m`, espacio y vamos a agregar un mensaje. Ese mensaje se pone entre comillas. ¿Pero qué mensaje podemos colocar acá?

[03:10] Es buena práctica poner un mensaje descriptivo pero que no sea grande. No vamos a escribir una documentación del archivo acá. Entonces podemos poner por ejemplo: "Creando archivo `index.html` con lista de cursos". Es lo suficientemente descriptivo para saber lo que ocurrió en ese archivo en ese punto del tiempo.

[03:38] Cuando dan "Enter", en este caso yo recibí este mensaje, que todo salió bien. Antes de entrar a este mensaje es posible que ustedes al apretar enter, ustedes hayan recibido un mensaje diciendo que las configuraciones de Git no están correctas o que falta configurar. En ese caso, lo que ustedes necesitan hacer es colocar: `git config`, espacio, `--global`, espacio, `user.email`, espacio, abren comillas y colocan su email.

[04:21] Pueden colocar el email de ustedes, voy a colocar aquí un email. Si usan la tecla con la flechita para arriba dentro de su teclado, ustedes pueden ver el mensaje anterior y pueden modificarlo y pueden colocar: `user.name`, espacio, y acá su nombre. En este caso voy a colocar Bruno y damos enter.

[04:53] Todo esto que les expliqué, solo si les aparece un mensaje diferente a esto que yo tengo acá, muy diferente. Si les ha salido un mensaje por el estilo, entonces no se preocupen, no es necesario hacer esto, porque ustedes ya tienen configurado Git. En este caso, lo que nos indican estos mensajes es primero, master que es el commit root, que dice acá o el commit base y nos está diciendo que este master es la rama.

[05:33] Esa rama nosotros ya lo vamos a ver más adelante. Y este de acá es el mensaje que nosotros colocamos anteriormente. Además, muestra las modificaciones. Nos está diciendo que un archivo fue modificado y además que 15 líneas fueron agregadas. Git tiene un monitoreo de línea por línea, dentro de cada archivo puede ver qué es lo que ha ocurrido línea por línea.

[06:03] Si ejecutamos git status nos dice que estamos en la branch master, eso no importa y que no hay nada para commitear. Dice que no hay nada para commitear pero no significa que no existan commits ahí dentro. Git ya sabe que hay un repositorio de Git, sabe que existe commits y no tengo ninguna modificación.

[06:29] Vamos a hacer una modificación. Dentro de nuestro archivo HTML, yo veo que acá por ejemplo faltó un acento. Entonces vamos a modificarlo. Le doy a modificar, voy a agregar el acento, voy a salvar el archivo haciendo clic en este caso, dentro de Visual Studio Code, que estoy usando. File, save y con eso guardamos el archivo o simplemente apretando "Ctrl + S".

[06:59] Vemos que se forma una bolita que había ahí y eso significa que ya fue guardado. Habiendo guardado nuestro archivo, nosotros lo que vamos a hacer ahora es colocar git status para ver qué es lo que está viendo Git. Colocamos eso y vemos acá que nos indica que fue modificado un archivo.

[07:27] Entonces, para agregar esa modificación hacemos git add, espacio y el nombre del archivo. O pueden colocar punto si solamente tienen este archivo

también dentro de su carpeta o si quieren agregar todos los archivos que existan ahí adentro. Bien. Damos enter.

[07:49] Si damos git status de nuevo nos dice que fue agregado ese index.html y que está listo para ser commiteado. Ahora ponemos git commit, espacio, -m, espacio, abro comillas, "Arreglado acento en el curso de integración continua". Bien. Pueden colocar acentos y símbolos dentro de ese mensaje.

[08:32] Damos enter y bien, ya hemos hecho nuestro commit, nos dice que un archivo fue modificado y se agregó una línea y se eliminó una línea. Esa línea que se agregó y esa línea que se modificó es la misma línea. Git lo toma como que se eliminó una línea y agregamos una nueva encima.

[08:56] Ahora, ¿cómo sabemos que se grabó realmente ese commit? ¿Existe una lista de commits hechos? Sí, existe. Lo podemos ver con un comando muy simple y lo vamos a ver en el próximo video.